融合社交网络与关键用户的并行协同过滤推荐算法 *

肖成龙，王　宁†，王永贵

(辽宁工程技术大学 软件学院, 辽宁 葫芦岛 125105)

摘　要：为解决传统协同过滤推荐算法中存在的数据稀疏、冷启动以及推荐结果缺乏多样性等问题，提出一种融合社交网络与关键用户的协同过滤推荐算法。该算法在用户-项目评分矩阵基础上，融合用户社交网络信息得出社交信任矩阵，融合关键用户信息得出关键用户评分矩阵。利用三大评分矩阵，分配不同的权重比例，共同来预测用户对于目标项目评分。针对海量数据问题，采用 Spark 分布式集群实现该算法的计算并行化。实验结果表明，该算法能够有效缓解数据稀疏问题，提高处理速度和推荐准确度。

关键词：社交网络；并行化；关键用户；协同过滤；大数据；电影推荐

中图分类号：TP301.6　　doi: 10.3969/j.issn.1001-3695.2018.04.0245

Parallel collaborative filtering recommendation algorithm based on social networks and key users

Xiao Chenglong, Wang Ning†, Wang Yonggui

(*College of Software*, *Laoning Program Technology University*, *Huludao Liaoning* 125105, *China*)

Abstract: In order to solve the problems such as sparse data, cold start and lack of diversity of recommendation results in traditional collaborative filtering recommendation algorithms, this paper proposes a collaborative filtering recommendation algorithm that integrates social networking with key users. Based on the score matrix of user projects, the algorithm integrates user social networks to derive social trust matrix, integrates key user information to obtain key user scoring matrix, and then uses these three matrix data distributions to predict user's target project with different weights. score. At the same time, aiming at the massive data problem, this paper uses the Spark distributed cluster to realize the parallelization of the algorithm. The experimental results show that the algorithm can effectively alleviate the data sparse problem and improve the data processing speed and recommendation accuracy.

Key words: social networking; parallelization; key users; collaborative filtering; big data; movie recommendation

0　引言

随着互联网行业的飞速发展，信息爆炸式增长，带来“信息过载”问题[1]，而有效的解决方法之一就是推荐系统。它可以根据用户的信息需求、兴趣爱好等，将用户感兴趣的内容或者产品推荐给用户，避免用户接收大量无用信息的可能，实现个性化的信息推荐。与搜索引擎提供的“一对多”式的信息服务不同，推荐系统输出的结果更符合用户的个性化需求，实现“一对一”式的信息服务，同时用户的参与程度也更低，从而极大降低用户搜寻信息的成本[2]。众所周知，推荐系统在日常生活中、各大门户网站中得到广泛的应用。

传统的推荐系统技术主要包括基于关联规则推荐(association rules)[3]、基于内容的推荐(content-based recommendation)[4]、协同过滤推荐(collaborative filtering)[2]和混合推荐(hybrid approach)[5]。其中基于关联规则推荐是以关联规则为基础，把已购商品作为规则头，规则体为推荐对象。简单的说就是在一个交易数据库中统计购买商品集 X 的交易中有多大比例的交易同时购买商品集 Y，其直观的意义就是用户在购买某些商品的时候有多大倾向去购买另外一些商品。但是对于关联规则的发现最为关键且耗时，是算法的瓶颈。基于内容的推荐，它是建立在项目的内容信息上作出推荐，而不需要依据用户对项目的评分，更多地需要用机器学习方法从关于项目内容的特征描述中得到用户的兴趣资料。但是它缺点是要求项目内容要具有结构化，同时该内容能够容易抽取成有意义的特征

收稿日期：2018-04-04；**修回日期：**2018-06-06　**基金项目：**国家自然科学基金资助项目（61404069）；辽宁省教育厅一般科研项目（LJYL048）；辽宁省科技厅博士启动基金项目（20141140）

作者简介：肖成龙（1984-），男，湖南祁阳人，副教授，博士，主要研究方向为软硬件协同设计、高层次综合、可扩展处理器；王宁（1993-），女（通信作者），硕士研究生，主要研究方向为数据挖掘、协同过滤推荐、约束规划（2226194809@qq.com）；王永贵（1967-），男，教授，硕士，主要研究方向为大数据分析、云计算、机器学习等.

表示。协同过滤推荐是推荐系统中应用最早和最为成功的技术之一。它一般采用最近邻技术，利用用户的历史喜好信息计算用户之间的相似性，然后利用目标用户的最近邻居用户集对商品的加权评价值来预测目标用户对特定商品的喜好程度，系统从而根据这一喜好程度来对目标用户进行推荐。但是协同过滤也存在一定的弊端，如冷启动问题，数据稀疏问题等。混合推荐系统则是根据不同推荐协同有不同的优缺点，将不同推荐方法进行组合，以扬长避短，然而尽管从理论上有很多种推荐组合方法，但在某一具体问题中并不见得都有效，组合推荐一个最重要原则就是通过组合后要能避免或弥补各自推荐技术的弱点。

相比于这几大类推荐技术，基于协同过滤算法的推荐系统具有较好的准确性、有效性，同时以其出色的鲁棒性和健壮性，在各大推荐系统中被广泛使用。基于此，协同过滤方法中存在的冷启动问题、数据稀疏问题也得到国内外学者的关注。冷亚军等人[6]认为数据稀疏性问题会从近邻搜寻不够准确和近邻评分过少两方面对协同过滤产生不利影响。翁小兰等人[7]也在综述中提出数据稀疏、冷启动和扩展性是协同过滤推荐算法所面临的主要问题。为解决数据稀疏问题，不同文献提出不同的解决方案，如文献[8]使用奇异值分解来降低评分矩阵的维数从而达到降低矩阵稀疏性的目的。孔欣欣等人[9]提出一种标签权重的评分方法来最大化地降低客观因素对用户评分的影响，有效缓解用户的评分偏差问题。同时很多学者提出信任关系之于协同过滤算法的必要性，如 Jøsang 等人[10]对能够获得信任度和声誉的系统进行综述，认为信任度和声誉应当作为安全机制应用于协同过滤系统中。文献[11]融合用户社交信任度和评分相似性，提出了一个新矩阵填充的推荐方法，使预测评分准确度明显提升，改善了推荐过程中存在的稀疏性问题。Yang 等人[12]提出一种结合用户信任关系的改进协同过滤算法，通过集成用户提供的稀疏评估数据和稀疏的社交信任网络数据来提高协同过滤的推荐性能。以上方法中均在不同程度有效缓解数据稀疏问题，提高推荐准确度，但是也存在各自的局限性，如奇异值矩阵分解计算复杂度高，存在过度拟合和缺乏精确性；对于只采用用户项目评分或者只采用用户信任关系数据，都会存在数据源单一问题，造成推荐结果失真[13]等。

因此，本文提出了一种融合社交网络与关键用户的并行协同过滤推荐算法。一方面使用 Spark 分布式集群提高对于海量数据的处理速度，另一方面同时考虑用户社交网络信息、用户群体中的关键用户数据信息和用户项目评分信息，有效的避免数据源单一问题，从而提高预测评分的准确性，有效的解决冷启动问题，提供具有多样性的推荐结果。本文算法主要包括以下几个步骤：a)数据预处理，对 Epinions 公开数据集进行数据分析，得到用户项目评分信息，用户信任社交网络信息以及用户群体中信任程度高的若干关键用户信息;b)计算相似性，将多数据源分配不同的权重配比，采用皮尔逊相关系数(Pearson correlation coefficient）计算用户之间的相似性，得到目标用户的近邻集合;c)预测评分，计算所有项目对目标用户的推荐度，选出 top$N$ 项目推荐给用户;d)准确性度量，统计准确度和决策支持准确度得出基于本文方法的推荐系统推荐质量。

1　传统协同过滤推荐算法简介

传统的协同过滤推荐算法主要包括基于用户的协同过滤推荐算法、基于项目的协同过滤推荐算法和基于模型的协同过滤推荐算法。三种推荐方法侧重点不同，适用环境也不同，但对于推荐系统来说一般包括以下三个步骤：首先是建立评分矩阵，然后是计算相似性，最后是预测评分得出推荐项目。推荐算法的每一个步骤中都包括一系列的计算和对数据的处理，具体过程介绍如下。

1.1　构建评分矩阵

建立用户-项目评分矩阵，将原始数据预处理得到如表 1 所示的用户评分矩阵 R，其中行代表用户(U)，列代表项目（I）。假设用户数量有 3 名，项目数量有 5 个,矩阵中的数值代表用户对项目的评分，分值 1-5 之间，分值越高代表该用户对该项目越感兴趣，若数值空着则代表该用户对该项目没有评分。如表 1 所示。

表 1　用户-项目评分矩阵

	I_1	I_2	I_3	I_4	I_5
U_1	3		4	4	5
U_2	5	2	2	3	2
U_3	3	4	4	4	5

1.2　计算相似性

相似性计算的准确性是影响推荐质量的重要因素。使用较为广泛的相似度计算方法主要包括欧式距离相似度、Jaccard 相似度、余弦相似度、修正的余弦相似度、Pearson 相似度[14,15]等。其中 Pearson 相似度方法适用于数据稀疏性小，用户共同评价的项目数量多的情况，对于本文中算法同时采用多个数据源，弥补了数据稀疏性问题，所以选用 Pearson 相似度，计算公式如式（1）所示。

$$sim(u,v)=\frac{\sum_{i\in I_{uv}}(r_{ui}-\overline{r_u})\times(r_{vi}-\overline{r_v})}{\sqrt{\sum_{i\in I_{ux}}(r_{ui}-\overline{r_u})^2}\sqrt{\sum_{i\in I_{uv}}(r_{vi}-r_v)^2}} \tag{1}$$

其中：r_{ui}，r_{vi} 分别代表用户 u 和用户 v 对项目 i 的评分，$\overline{r_u}$ 代表用户 u 对项目的平均分，$\overline{r_v}$ 代表用户 v 的平均分，I_{uv} 代表户 u 和用户 v 评分过的项目交集，$sim(u,v)$ 代表求得的用户 u 与用户 v 之间的相似性。

1.3　预测评分及给出推荐

相似性计算结束后选取若干个相似度高的用户，作为目标用户的近邻集合，由近邻用户的评分数据进行计算，预测目标用户对项目的评分，得出评分较好的 Top-N 个推荐项目，为用户进行推荐。

要预测用户 u 对项目 i 的评分，需要考虑目标用户的邻居集合以及邻居集对项目的评分情况，令 N_u 为用户 u 的最近邻

居集合,则预测评分公式如式（2）所示。

$$P_{ui} = \overline{r_u} + \frac{\sum_{v \in N_u} sim(u,v) \times (r_{vi} - \overline{r_v})}{\sum_{v \in N_u} |sim(u,v)|} \tag{2}$$

经计算，得出项目预测评分后，选取其中的评分较高的项目(topN)推荐给用户，完成推荐功能。

1.4　协同过滤一般流程

本章节内容主要介绍了协同过滤推荐系统的一般流程，从原始数据到评分矩阵，计算相似性和预测评分，给出推荐结果。流程如图 1 所示。

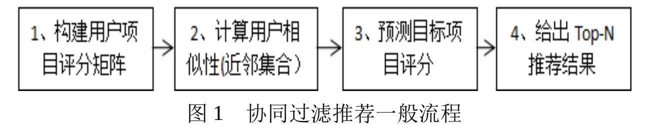


图 1　协同过滤推荐一般流程

2　融合社交网络与关键用户的协同过滤推荐算法

与传统协同过滤推荐算法不同，本文提出融合用户社交网络信息和关键用户数据的并行协同过滤推荐算法，扩展了数据源，将单一数据源增加为三个数据源，有效的缓解数据集稀疏问题，三个数据源配比不同的权重，提高预测的准确性、推荐的多样性。同时提出寻找用户群中的关键用户策略，关键用户具有较高的信任度，影响力，更具有普适性。适用于解决协同过滤推荐系统中的冷启动问题，对于新用户，缺乏评分数据以及社交信任关系，此时可采用关键用户信息数据为新用户进行系统推荐。

2.1　算法描述与模型设计

本文算法主要包括如下几部分：首先是数据预处理，包建立用户项目评分矩阵，建立用户社交评分矩阵，寻找关键用户并建立关键用户评分矩阵。然后使用数学优化分析综合工具软件 1stOpt（first optimization）进行回归分析得出三个数据源的对应权重参数，同时对数据进行归一化处理，再计算相似度，得出目标用户相似度高的近邻集合，最后由近邻集合对项目评分情况得出目标项目的预测评分，选取 Top-N 个推荐项目。

2.1.1 构建社交网络信任矩阵

对于不同的数据集，提取社交网络信任矩阵的方式不同。对于本文中采用的是已知信任关系是公开数据集 Epinions，其中用户之间的信任关系显示表示，信任数据范围是[0,1]，无评分则为空。对于没有显性信任数据时，信任度可以依据用户共同评分项等隐性信任数据构建，即用户之间的共同评分项的评分趋向越一致，则用户之间的信任程度越高，反之越低，如 UPS（user position similarity）方法[16]、用户间接可信度[17]等方法构建用户之间的信任度。用户信任关系如表 2 所示。

表 2　用户信任关系矩阵

	U_1	U_2	U_3
U_1	×	1	1
U_2	0	×	1
U_3	1	1	×

其中用户之间的信任关系是双向的，即用户 a 信任用户 b，但是用户 b 不一定信任用户 a。同时，对于非显示性信任数据，用户之间的信任程度也不一致，比如用户 c 对用户 d 的信任程度为 0.9，而用户 d 对用户 c 的信任程度为 0.2，对于非显示性信任数据需要进行归一化处理。

2.1.2 构建关键用户评分矩阵

所谓关键用户是指具有较高的信任度，较多的评分记录，影响力高的用户。在数据集所有用户数据中，找出信任度高的关键用户，提取关键用户的评分数据，得到关键用户的评分矩阵 T。对于关键用户的数量和质量，可根据具体实验数据进行设置。截取 Epinions 数据集部分数据（前 100 条数据），通过数据可视化软件 Gephi 处理，得到如图二所示的信任关系。可知，对于大量数据中，均会存在信任关系较为集中的数据点，该点对应的用户就是该数据集中的关键用户。如图 2 中编号为 2824 的节点，编号为 4294 的节点和编号为 23298 的节点，相比于数据集中其他节点，关键用户节点的度数较多，因此较为重要，具体可类比网页排名 PageRank 算法[18]。

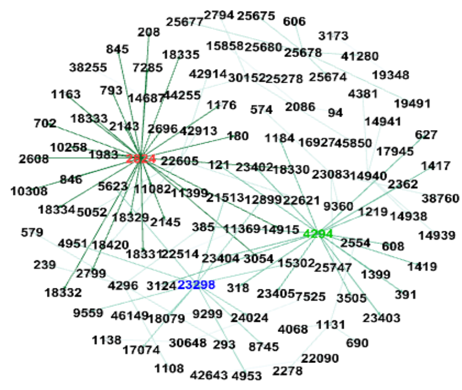


图 2　社交网络中的关键用户

根据信任度数，设置阈值，选取关键用户并提取关键用户评分矩阵如表 3 所示，其中 U_k 代表关键用户。

表 3　关键用户项目评分矩阵

	I_1	I_2	I_3	I_4	I_5
U_{k1}	5	3	4	4	5
U_{k2}	5	3	4	4	4
U_{k3}	4	4	4	4	5

2.1.3 相关相似性计算

对于本文中提出的算法，除需要计算用户项目评分矩阵对应的用户相似性外，还需要计算社交信任关系矩阵和关键用户评分矩阵分别对应的相似性。相似性计算方法与公式（1）相同，得到不同矩阵对应的相似性结果后，通过不同的权重配比得出最终相似性计算公式如（3）所示。

$$\begin{aligned} sim(u,v) &= \alpha user_sim(u,v) + \\ &\beta sns_sim(u,v) + \gamma key_sim(u,v) \end{aligned} \tag{3}$$

其中：α，β，$\gamma \in (0,1)$ 而且 $\alpha + \beta + \gamma = 1$，它们是对不同评分矩阵的相似性调控参数。$user_sim(u,v)$ 是用户项目评分矩阵计算的用户相似度，$sns_sim(u,v)$ 是社交网络信任矩阵得到的用户相似度，$key_sim(u,v)$ 是基于关键用户评分矩阵得到的用户相

似度，由此得出最终的用户相似度 $sim(u,v)$。

2.1.4 算法流程

本文提出的融合社交网络与关键用户的协同过滤推荐算法主要包括如下几个步骤：a)数据预处理由原始数据集得到三大数据矩阵，具体过程如上述所示;b）采用皮尔逊相关系数计算项目相似度，公式如式（3）所示;c）由邻居集项目评分预测目标项目的评分，公式如式（2）所示。最后得到 topN 个推荐项目。整个算法流程如图 3 所示。

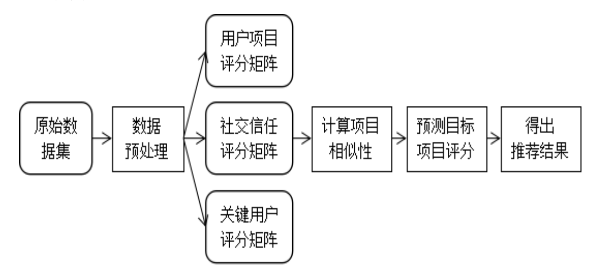


图 3　本文算法流程示意图

2.2　并行化算法流程设计

在 Spark 分布式集群系统中实现融合社交网络与关键用户的协同过滤推荐算法，经过数据预处理后，得到三大矩阵数据，在 Spark 中为矩阵提供了不同方式的分布式存储选择和多种矩阵计算操作。Spark 编程模型中提供了各种算子操作，比如 map(),reduce(),join(),filter()等。通过对三大矩阵的计算得出三个相似度 RDD，之后以不同的权重配比合并三个相似度 RDD，得出最终的用户间的相似度，计算预测评分，给用户推荐项目。在 Spark 实现并行化过程中，数据均以 RDD 格式存储，RDD 数据集分布式的存储在 Spark 内存中，整个过程中只需要访问一次数据存储文件系统 HDFS，相比于 hadoop 集群，减少数据访存次数，降低通信开销。算法并行化实现的伪代码如下所示：

本文算法 Spark 并行化实现

```
Input ： userid::movieid::rating; userid::userid::trust
Output ： 推荐给用户 N 个感兴趣的电影

#从 HDFS 中读取评分数据、信任关系和关键用户评分
Sc1 = SparkContext(sys.argv[1],"Userbased");
Sc2 = SparkContext(sys.argv[2],"SNSbased");
Sc3 = SparkContext(sys.argv[3],"Keyuserbased");
Lines1 = Sc1.textFile(sus.argv[4]);
Lines2 = Sc2.textFile(sus.argv[4]);
Lines3 = Sc3.textFile(sus.argv[4]);
Partions = P;
#并行化实现，得到用户-项目倒排表（以 Lines1 为例）
Item_user = Lines1.parallenlize(0 until P).map(parseVectorOnItem).groupByKey().map(x[0].
x[1].500).cache()
#统计用户间都喜欢的物品
Pairwise_users = Item_user.filter(lambda x:len(x[1]>1)).map(lambda x:findUserParirs(x[0],x[1])).groupByKey()
#计算相似度，包括用户相似度（user_sim(u,v)），与信任用户相似度（sns_sim(u,v)）和与关键用户相似度（key_sim(u,v)），以 user_sim(u,v)为例
user_sim(u,v) = Pairwise_users.map(lambda x:calcSim(x[0],x[1])).
map(lambda x:keyOnFirstUser(x[0],
x[1])).groupByKey().map(lambda x:nearestNeighbors(x[0],x[1],50));
#组合相似度，配比不同权重得到最终相似度
sim(u,v) = user_sim(u,v).join(sns_sim(u,v)).join(key_sim(u,v)).map(x => αx[1]+βx[2]+γx[3] ),sortKey().cellect
#为用户推荐前 N 个感兴趣的物品
Recommend_item = sim(u,v).map(lambda x:topNRecommendations(x[0],x[1],uib.value,50)).collect
```

3　实验及结果分析

3.1　实验数据

本文实验数据选择 Epinions 公开数据集作为实验数据，它是由 Massa 在 http://www.epinions.com 网站收集整理所[19]。共包括 49,290 位用户评分对 139,738 个不同的项目评分，评分数据为[1,5]之间的整数，得分越高表示用户越喜欢该项目。同时也包含 664 824 条用户间的社交数据，其中 487 181 条记录表示用户间的关系是积极的，认为是信任数据，信任值为 1，其余不存在信任关系即为 0。

3.2　实验环境

本文中实验集群使用的是青云控制台(https://www.qingcloud.com/)提供的专业 Spark 集群，免去集群搭建的繁琐过程，同时严格控制集群中每一台机器的完整一致性，提供广泛的型号规格可选择空间，包括不同处理器型号数据，大小不等的内存配置主机。本实验中设置一个主节点（master）和五个从节点（worker），Spark 版本为 2.2.0，六台主机均为双核处理器，4GB 运行内存。

3.3　测评指标

常用的协同过滤推荐算法评价指标，主要包括两大类，一类是评分准确度，通过计算平均绝对误差(MAE)、平均平方根误差(MSE)、均方根误差(RMSE)和标准平均绝对误差(NMAE)对推荐系统进行评价，适用于比较关注精确的预测评分系统;另一类是分类准确度，通过计算准确率 (precision) 、召回率(recall)、F 1 指标和 ROC 曲线面积来评价推荐系统，适用于

有明确二分喜好的系统[20]。其中，准确率定义为系统的推荐列表中用户喜欢的产品和所有被推荐产品的比率；准确率表示用户对一个被推荐产品感兴趣的可能性。召回率定义为推荐列表中用户喜欢的产品与系统中用户喜欢的所有产品的比率；召回率表示一个用户喜欢的产品被推荐的概率。除此两大类外，还有计算评分关联信息、计算排序加权指标和覆盖率等评价指标。在本文中采评分准确度中的平均绝对误差(MAE)和均方根误差(RMSE)；分类准确度中的准确率 (precision)和召回率(recall)进行评价。

评分准确度中 MAE 计算的是所有训练集测试用户对测试项目的预测评分和实际评分的平均误差大小，计算公式如式（4）所示，其中 T_u 表示训练集中的用户数据集合； N 表示训练集项目数目； $P_{u,i}$ 表示用户 u 对项目 i 的预测评分； $R_{u,i}$ 表示用户的真实评分。

$$MAE = \frac{1}{N}\sum_{u \in T_u} |P_{u,i} - R_{u,i}| \tag{4}$$

RMSE 计算用户的真实评分与预测评分值的均方根误差，计算公式如式（5）所示。

$$RMSE = \sqrt{\frac{1}{N}\sum_{u \in T_u} (P_{u,i} - R_{u,i})^2} \tag{5}$$

由于本文数据集为 5 分制数据，欲将其划分为二分制，则认为评分大于 3 的商品认为是用户喜欢，反之认为用户不喜欢。公式中 $R(u)$ 代表为用户推荐的项目列表，T_u 为测试集中用户评分高于 3 分的项目列表。分类准确度准确率计算公式如式（6）所示。

$$\Pr ecision = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|} \tag{6}$$

召回率计算公式如式（7）所示。

$$\operatorname{Re} call = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|} \tag{7}$$

3.4 实验结果分析

为了评价本文算法的综合性能，将本文算法与其他算法进行如下几组对比实验。

a)传统基于用户的协同过滤推荐系统（UserCF）。传统算法是仅使用用户-项目评分矩阵，通过计算用户间的相似度进行预测评分和推荐。

b)传统基于项目的协同过滤推荐系统（ItemCF）。该方法同样使用用户-项目评分矩阵，但计算方式与 UserCF 不同，计算项目间的相似度进行预测评分和推荐。

c)融合社交网络特征的协同过滤推荐算法(TDSRec)[21]。该方法在考虑社交网络的同时，融合基于用户评分偏好的相似性，共同对用户评分矩阵中的数据值进行评分预测。

c)融合社交网络与关键用户的协同过滤推荐算法（SNKUCF）。本文算法同时兼顾用户项目评分矩阵，用户社交网络信任矩阵和关键用户评分矩阵，以不同的相似度权重共同来预测评分，得出推荐结果。

根据以上算法，使用同一公开数据集，同一主机分别测试本文算法和其他算法，计算对应的评价指标，得到如图 4 所示结果。

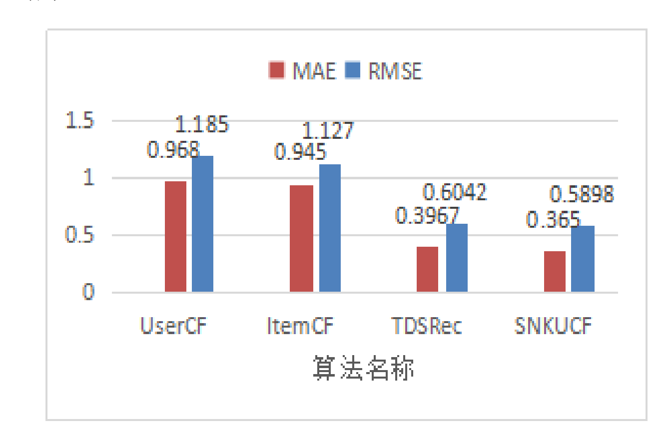


图 4　四种算法实验结果对比

在数据集 Epinions 中选取前 500 条数据，测试针对三大相似度配比不同权重的效果差异，其中已知 $\alpha+\beta+\gamma=1$，且 $0<\alpha,\beta,\gamma<1$，分别设置不同取值，共测试 42 种权重分配情况，其中当 $\alpha=\beta=0.4$, $\gamma=0.2$ 时，效果最好。由该权重配比，测试本文算法与其他算法的平均绝对误差和均方根误差。同时测试了该权重配比下，对于不同规模的数据集，其准确率和召回率变化情况，分别选取数据集前 100 条数据，前 500 条数据，前 1000 条数据等进行测试，实验结果如图 5 所示。

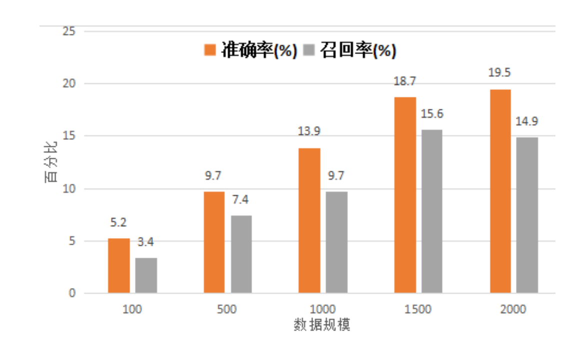


图 5　准确率和召回率

由图 5 数据可知，随着数据规模的增大，预测准确率在逐步增加，成正相关。与此同时，召回率也在逐步增加，但在数据集为 2000 条数据时有回落趋势。在其他算法中准确率与召回率基本成负相关关系，本文算法与其他实验结果趋势相符，但趋势变化范围没有其他算法明显，在本文算法中用户项目评分和社交网络信任用户评分可以有效提高推荐的准确度，引入关键用户评分则有效扩展推荐的多样性，因为关键用户之间兴趣差异度较大。

同时测试了在 Spark 集群中共 6 台主机，本文算法的时间效率，与单机算法运行时间进行对比，计算集群加速比。加速比，是同一个任务在单处理器系统和并行处理器系统中运行消耗的时间的比率，用来衡量并行系统或程序并行化的性能和效果。计算公式如式(8)所示。

$$S_P = \frac{T_1}{T_P} \tag{8}$$

其中：S_P 是加速比，T_1 是单处理器下的运行时间，T_P 是在有 P 个处理器下的运行时间。实验结果如表 4 所示。

表 4 并行化性能验证

数据规模	集群规模	运行时间/min	加速比
10000 条	1	3.56	_
	4	1.02	3.39
	6	0.65	5.48
20000 条	1	7.08	_
	4	1.93	3.67
	6	1.20	5.89
30000 条	1	14.24	_
	4	3.68	3.86
	6	2.71	5.25
40000 条	2	16.72	_
	4	4.41	3.79
	6	2.84	5.88

分布式集群计算结果表明，并行化处理能够有效提高问题求解速度，集群规模与时间效率基本上成线性负相关，同时数据分配过程中不可避免会产生通信代价，消耗部分时间，但由加速比变化情况可知随着数据规模的增加，通信代价所占比例逐步减小。

4 结束语

为解决推荐系统中存在的评分数据稀疏，冷启动问题，以及多样性问题，对于大数据的处理问题，本文提出并实现了融合社交网络和关键用户的并行协同过滤推荐算法，由本文算法实验结果表明，该算法能够有效缓解数据稀疏问题，提高推荐准确性，缩短系统运行处理时间。

未来拟对推荐系统中评价指标进行进一步研究，以提出其他评价方法，用来全面系统评价推荐质量，包括对推荐结果的多样性，新颖性等指标进行量化评价。

参考文献：

[1] 刘红霞. 基于协同过滤技术的推荐系统综述 [J]. 信息安全与技术, 2016, 7 (3) . (Liu Hongxia. A Survey of Collaborative Filtering Technique in Recommendation System [J]. Information Security and Technology, 2016, 7 (3)

[2] 夏培勇. 个性化推荐技术中的协同过滤算法研究. 博士学位论文. 青岛: 中国海洋大学, 2011. (Xia Peiyong. Research on Collaborative Filtering Algorithm of Personalized Recommendation Technology [D]. QingDao: Ocean University of China, 2011)

[3] 苏新宁, 杨建林, 邓三鸿, 等. 数据挖掘理论与技术. 北京: 科学技术文献出版社, 2003. (Su Xingning, Yang Jianlin, Deng Sanhong, *et al.* Data Mining Theory and Technology [M]. Beijing: Science and Technology Literature Publishing House, 2003)

[4] BaezaYates, Ricardo A, RibeiroNeto, *et al.* Modern Information Retrieval [J]. 1999, 43 (1): 26-28.

[5] 许海玲, 吴潇, 李晓东, 等. 互联网推荐系统比较研究 [J]. 软件学报, 2009, 20 (2): 350-362. (Xu Hailing, Wu Xiao, Li Xiaodong, *et al.* Comparison study of internet recommendation system [J]. Journal of Software, 2009, 20 (2)

[6] 冷亚军, 陆青, 梁昌勇. 协同过滤推荐技术综述 [J]. 模式识别与人工智能, 2014, 27 (8): 720-734. (Leng Yajun, Lu Qing, Liang Changyong. Survey of Recommendation Based on Collaborative Filtering [J]. PR&AI, 2014, 27 (8): 720-734)

[7] 翁小兰, 王志坚. 协同过滤推荐算法研究进展 [J]. 计算机工程与应用, 2018, 54 (1): 25-31 (Weng Xiaolan, Wang Zhijian. Research process of collaborative filtering recommendation algorithm. Computer Engineering and Applications, 2018, 54 (1): 25-31)

[8] Kumar R, K. Verma B, Sunder Rastogi S. Social popularity based SVD+recommender system [J]. International Journal of Computer Applications, 2014, 87 (14): 33-37.

[9] 孔欣欣, 苏本昌, 王宏志, 等. 基于标签权重评分的推荐模型及算法研究 [J]. 计算机学报, 2017, 40 (6): 1440-1452. (Kong Xinxin, Su Benchang, Wang Hongzhi, *et al.* Research on the modeling and related algorithms of label-weight rating based recommendation system [J]. Chinese Journal Of Computers2017, 40 (6): 1440-1452)

[10] Jøsang A, Ismail R, Boyd C. A survey of trust and reputation systems for online service provision [J]. Decision Suppo-rt Systems, 2006, 43 (2): 618-644.

[11] Wang Meiling, Ma Jun. A novel recommendation approach based on users weighted trust relations and the rating similarities [J]. Soft Computing, 2015, 20 (10): 3981-3990.

[12] Yang B, Lei Y, Liu J, *et al.* Social collaborative filtering by trust [J]. IEEE Trans on Pattern Analysis & Machine Intelli-gence, 2017. 39 (8): 1633-1647.

[13] 王升升, 赵海燕, 陈庆奎, 等. 个性化推荐中的隐语义模型 [J]. 小型微型计算机系统, 2016, 37 (5): 881-889. (Wang Shengsheng, Zhao Haiyan, Chen Qingkui, *et al.* Latent Factor Model for Personalized Recommendation [J]. Journal of Chinese Computer Systems, 2016, 37 (5): 881-889)

[14] 刘青文. 基于协同过滤的推荐算法研究 [D]. 合肥: 中国科学技术大学, 2013. (Liu Qingwn. Research on Recommender Systems based on Collaborative Filtering [D]. Hefei: University of Science and Technology of China, 2013)

[15] 李斌. 推荐系统研究综述 [J]. 现代计算机, 2014 (2): 7-10. (

[16] Li Bin. A Survey of Recommender System [J]. Modern computer, 2014 (2): 7-10)

[17] Du Yongping, Du Xiaoyan, Huang Liang. Improve the collaborative filtering recommender system performance by trust network construction

[J]. Chinese Journal of Electronics, 2016, 25 (3): 418-423.

[18] Massa P, Bhattacharjee B. Using Trust in Recommender Systems: An Experimental Analysis [C]// Proc of the 2nd International Conference, Trust Management. 2004: 221-235.

[19] 孙红，左腾. 基于 PageRank 的微博用户影响力算法研究 [J]. 计算机应用研究, 2018, 35 (4) . (Sun Hong, Zuo Teng. Research on algorithm of micro-blog user influence based on PageRank [J]. Application Research of Computers, 2018, 35 (4)

[20] Rennie J D M, Srebro N. Fast maximum margin matrix factorization for collaborative prediction [C]// Proc of the 22nd International Conference on Machine Learning. New York: ACM, 2005: 713-719.

[21] 朱郁筱，吕琳媛. 推荐系统评价指标综述 [J]. 电子科技大学学报, 2012, 41 (2): 163-175. (Zhu Yuxiao, Lv Linyuan. Evaluation metrics for recommender systems [J]. Journal of University of Electronic Science and Technology of China, 2012, 41 (2): 163-175)

[22] 郭宁宁，王宝亮，侯永宏，等. 融合社交网络特征的协同过滤推荐算法 [J]. 计算机科学与探索, 2018, 12 (2): 208-217 (Guo Ningning, Wang Baoliang, Hou Yonghong, *et al.* Collaborative filtering recommendation algorithm based on characteristics of social network [J]. Journal of Frontiers of Computer Science and Technology, 2018, 12 (): 208-217) .